One powerful feature of a CI/CD pipeline is that we can have multiple steps in the build process, which means we can make modifications to the project before the actual build is run.

The Rubeus project includes a YARA rule file, which detects the default TypeLibGUID of the project: `658C8B7F-3664-4A95-9572-A3E5871DFC06`. This is hardcoded in the AssemblyInfo.cs file.

You can verify the YARA rule with the artifact from your first build.

```
PS C:\Users\Daniel\Downloads\yara-4.2.3-2029-win64> .\yara64.exe .\Rubeus.yar .\Rubeus.exe -s
HackTool_MSIL_Rubeus_1 .\Rubeus.exe
0x6b6ba:$typelibguid: 658c8b7f-3664-4a95-9572-a3e5871dfc06
```

We can add a build step to our pipeline to change this GUID prior to building the final assembly.

Back under the build configuration > build steps, click **add build step** and select the **PowerShell** runner (since PowerShell provides a very easy means of just changing a string in a file).

Under the **Script** dropdown, select **Source code**. Now we can enter the PowerShell code directly into this text box. We're going to use `Get-Content` to read the target file, replace the target GUID with a random one, then overwrite the file with the new content.

It's worth noting that during a build, the repository code is checked out to a different directory. So, we're only modifying the checked-out version, not the original source.

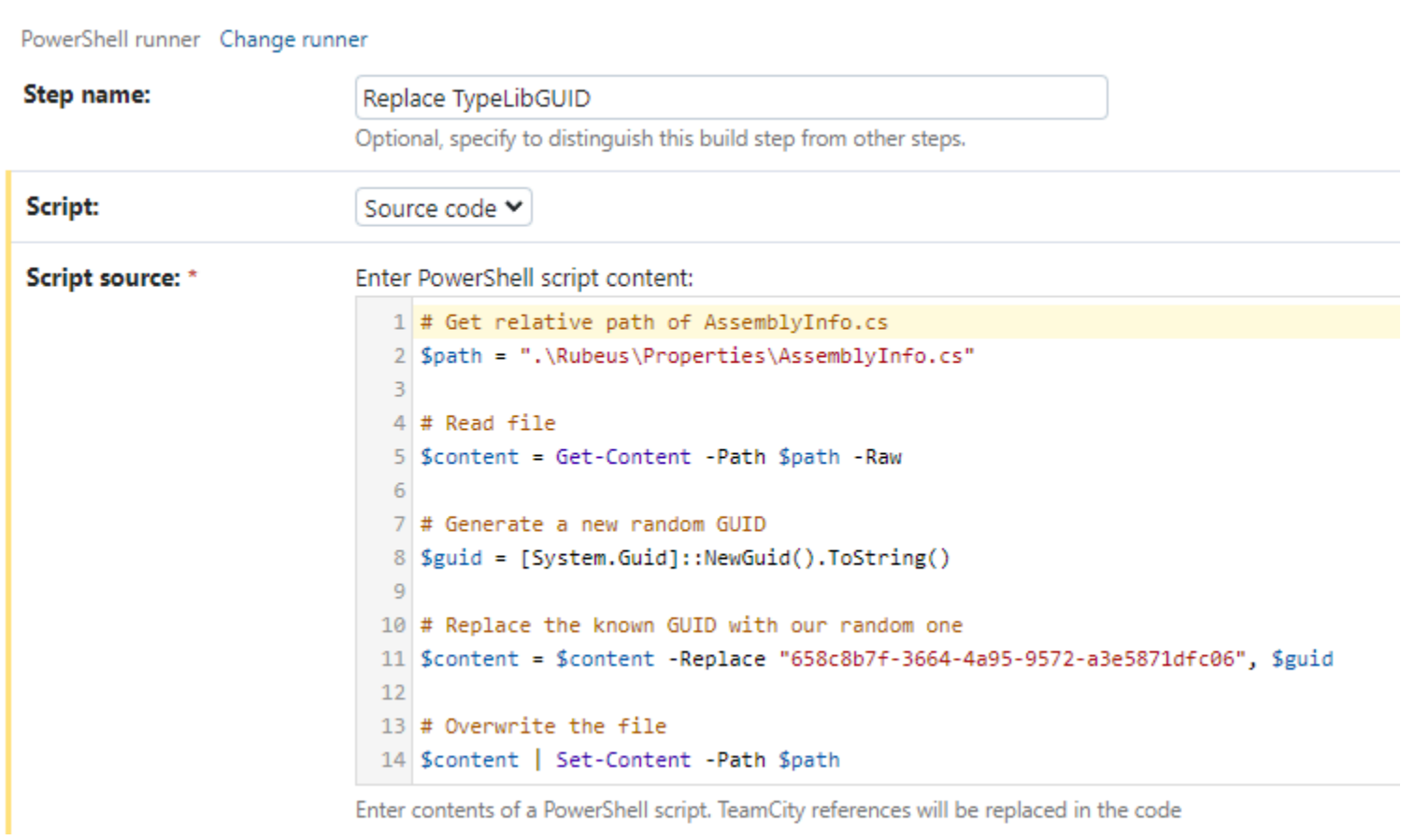
```
# Get relative path of AssemblyInfo.cs
$path = ".\Rubeus\Properties\AssemblyInfo.cs"

# Read file
$content = Get-Content -Path $path -Raw

# Generate a new random GUID
$guid = [System.Guid]::NewGuid().ToString()

# Replace the known GUID with our random one
$content = $content -Replace "658c8b7f-3664-4a95-9572-a3e5871dfc06", $guid

# Overwrite the file
$content | Set-Content -Path $path
```



Click **Save** and you'll be taken back to the list of build steps. New build steps are automatically added to the bottom of the list, simply click **reorder build steps** and drag the new step to the top.

After running the build pipeline again and downloading the new artifact, the YARA rule will no longer match.

```
PS C:\Users\Daniel\Downloads\yara-4.2.3-2029-win64> .\yara64.exe .\Rubeus.yar .\Rubeus.exe -s
PS C:\Users\Daniel\Downloads\yara-4.2.3-2029-win64>
```

Replacing strings is something that can be expanded in other projects, such as PowerView. We could remove comments, change variable names, and much more. And since we're generating elements randomly at build time, each build will produce a unique artifact.